The Approximation of Implicates and Explanations

by

Alex Kean

Technical Report 90-31
January, 1991

Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1W5

The Approximation of Implicates and Explanations

Alex Kean

Department of Computer Science, University of British Columbia,
Vancouver, British Columbia V6T 1W5, Canada

Technical Report 90-31

December 1990

This paper studies the continuum between implicates and minimal implicates; and the continuum between explanations and minimally consistent explanations. The study is based on the approximation of the set of these objects. A general definition for approximated minimal implicates, called selective implicates, is presented. Three specific instances of selective implicates: *query-based*, ATMS and *length-based* are studied. Using the set of *query-based* minimal implicates, explanations are generated and the properties of these explanations are studied. The goal of these studies is to extract computationally feasible properties in order to achieve tractable abduction. The setting is the compiled approach using minimal implicates in *Clause Management Systems.*

1 Introduction

In Artificial Intelligence, *abduction* — a form of reasoning by finding explanations — is emerging as an important reasoning paradigm in many applications such as diagnostic reasoning. This is evident in the development of systems like RESIDUE [6], THEORIST [11], ATMS [3]; and theoretical work like Clause Management Systems [12, 13, 9].

In Clause Management Systems (CMS), abduction is formulated as deducing a consistent sentence (an explanation) that implies a given query from its knowledge base. The computation of abduction in the CMS is formulated in terms of compiling (preprocessing) its knowledge base into a set of equivalent sentences called minimal implicates. The search for an explanation for a given query is achieved by fast set operations on the set of minimal implicates. Unfortunately, this achievement is attained at the expense of the potentially exponential storage required for the set of minimal implicates [2, 8].

In light of such a drawback, a problem in computational abduction is to find a useful tradeoff between preprocessing and explanation finding in abduction. In this paper, we shall study this tradeoff by examining the role of approximation of implicates and explanations.

What is an approximation ? One possible definition is *in terms of the quality of being close or near the desired quality*. In the case of implicates, the desired quality is the property of being a minimal implicate of its knowledge base Σ. In the case of explanation, the desired quality is being the minimally consistent explanation of a given query with respect to Σ. The difference between the *desired quality* and *that of the approximation* will be the measure of the adequacy of the approximation.

1.1 Implicates and Explanations

We shall first examine the role of implicates in abduction. To simplify the presentation, we shall assume a propositional language $\mathcal{L}$ over a vocabulary $\mathcal{V}$. A *literal* is either x or $\neg x$ for any $x \in \mathcal{V}$, a *clause* C is a disjunction of literals, and a *formula* Σ is a conjunction of clauses. Clauses and formulae are sometimes represented by sets, thus if $C = c_1 \vee c_2 \vee c_3$, its corresponding set notation is $C = \{c_1, c_2, c_3\}$. The negation of a clause (or set) $\overline{C}$ (represented by the over-strike bar), is $\neg c_1 \wedge \neg c_2 \wedge \neg c_3$ and in set notation $\{\neg c_1, \neg c_2, \neg c_3\}$. For legibility, we shall use implication as in $a \wedge b \rightarrow c$ to denote the clause $\neg a \vee \neg b \vee c$. And, we shall also use $E \rightarrow G$ to denote $\overline{E} \cup G$ for clauses $\overline{E}$ and G.

A clause P *subsumes* a clause Q if $P \subseteq Q$. The function $SUB(\Sigma)$ is a subset of Σ such that no $C \in SUB(\Sigma)$ is subsumed by another clause $C' \neq C$ in Σ. A clause R is *fundamental* if a pair of *complimentary* literals $\{x, \neg x\} \not\subseteq R$ for any x, otherwise it is *non-fundamental*.

Given a formula Σ, a clause P such that $\Sigma \models P$ is an implicate of Σ. An implicate P of Σ is *minimal* if there is no other implicate $P' \neq P$ of Σ such that P' subsumes P. In more familiar terms, the set of all implicates of Σ is the set of all theorems of Σ (i.e $Th(\Sigma)$). Thus, the set of all minimal implicates of Σ is the set

$$MI(\Sigma) = \{M \mid \text{ no } M' \not\subset M \text{ for } M, M' \in Th(\Sigma)\}.$$

Studying the set of minimal implicates is of great importance in the computational aspect of abduction. In abduction, an explanation for G with respect to Σ is defined as $(1)\Sigma \models E \rightarrow G$ and $(2)\Sigma \not\models \overline{E}$ (or $\Sigma \cup E$ is consistent). Similarly, a minimal explanation E of G with respect to Σ is one such that no proper subset of E is an explanation for G with respect to Σ.

If we knew the set $MI(\Sigma)$, finding a minimal implicate $E \rightarrow G \in MI(\Sigma)$ will ensure that E is a valid explanation for G and is consistent with Σ. For example, let $\Sigma = \{a \wedge b \rightarrow g, \overline{a}\}$. The clause $a \wedge b$ cannot be an explanation for g because it is inconsistent with $\overline{a}$. By minimality, $\overline{a} \in MI(\Sigma)$ ensuring that the clause $a \wedge b \rightarrow g$ is not in $MI(\Sigma)$ and therefore there is no explanation for g in Σ.

Additionally, the minimality of an implicate guarantees to some extent the minimality of the explanation[1]. For example, let $\Sigma = \{a \wedge b \rightarrow g, a \rightarrow g\}$. The explanation a for g is more minimal than $a \wedge b$. Again, by minimality, $\{a \rightarrow g\} \in MI(\Sigma)$ ensuring that the clause $a \wedge b \rightarrow G$ will not be considered. Thus, the minimality of an implicate plays two important roles in finding an explanation:

1. to ensure consistency of the explanation E, and

2. to achieve to some extent the minimality of E.

The approach of pre-computing minimal implicates to speed search for minimal explanations is extremely useful when the turn-around time for query processing is crucial. In another scenario where response time to query processing is compromized in favour of storage space, the interpreted approach supplemented with partial compilation is called for and is studied elsewhere [7].

In the case of the compiled approach unfortunately, the most distinct drawback is the size of the set of pre-computed minimal implicates. It is known in the literature that in the worst case the number of prime implicates (a subset of minimal implicates) of a formula Σ is exponential to Σ [2].

When the set of prime implicates is compiled incrementally that is, Π is a set of prime implicates of Σ and C is a new clause, the number of prime implicates of $\Pi \cup C$ is the size of Π with an exponent of the size of C [8].

1.2 Constraints and Approximation

In fact, the other drawback is that most of the minimal implicates compiled are not used in any way in the query processing. For illustration, consider a formula

$$\Sigma = \{a \rightarrow b,\ \ b \rightarrow c,\ \ c \rightarrow d,\ \ d \rightarrow g\}$$

where g is the query literal. The set of all minimal implicates of Σ are

$$\begin{aligned}
&\{\ a \rightarrow b, a \rightarrow c, a \rightarrow d, a \rightarrow g,\\
&b \rightarrow c, b \rightarrow d, b \rightarrow g,\\
&c \rightarrow d, c \rightarrow g,
\end{aligned}$$

[1]Unfortunately, it does not guarantee the absolute minimal explanation. To achieve that, the set of all explanations of G with respect to Σ must be available to ensure minimality. This is an inherent property of minimality [12, 13].

$d \rightarrow g,$

$a \rightarrow a, b \rightarrow b, c \rightarrow c, d \rightarrow d, g \rightarrow g\}.$

The set of all minimal explanations of g with respect to Σ is the set $\{a, b, c, d, g\}$. Notice that the set of so called "intermediate" clauses involved in the transitive closure for g, for instance $a \rightarrow c, a \rightarrow d, b \rightarrow d \ldots$, is not used to process the query g after its compilation. The following table illustrates those implicates that are kept and those that are ignored.

original	keep	forget
$a \rightarrow b$	$a \rightarrow g$	$a \rightarrow c$
$b \rightarrow c$	$b \rightarrow g$	$a \rightarrow d$
$c \rightarrow d$	$c \rightarrow g$	$b \rightarrow d$
$d \rightarrow g$		$\vdots$

Such scenario arises very commonly in circuit diagnosis where the query g is usually some observation of the inputs and outputs, and the "intermediate" clauses are those describing the circuits in between the input and the output.

This suggests that when using the compiled approach in a setting where the query literals are known, only the set of minimal implicates that are "relevant" to the query should be kept. If such scheme is employed, the obvious question would be what is the relationship between this restricted set of minimal implicates and the unrestricted minimal implicates ? The simple answer is that the restricted set is a subset of the set of minimal implicates, precisely corresponding with our motivation of storage saving and approximation. This suggests that a general scheme of implicate restriction can be defined for the purpose of abduction.

Definition 1.1 *A clause C is a* **selective** *implicate of Σ if $\Sigma \models C$ and $\mathcal{R}(C)$ holds for some constraint $\mathcal{R}$. The clause C is a* **selective minimal** *implicate of Σ if there is no other implicate $C' \neq C$ of Σ such that C' subsumes C.*

In general, a set of minimal implicates of Σ that is *selective* will be a subset of $Th(\Sigma)$ and also a subset of $MI(\Sigma)$. We can view this restricted set as a form of approximation of the minimal implicates of Σ. The differences between the minimal explanation for the query G with respect Σ obtained from the restricted set and the unrestricted set will be the measurement of success in the approximation. This subset may not be easily computed and hence an approximated computation is called for.

This paper studies such scheme of approximation of minimal implicates in the compiled approach for abduction. The goal is to discover the type of restriction on implicates that has overall saving in storage and minimal inaccuracy in the explanations computed.

2 Query-based Implicates

As suggested in the previous section, a useful restriction $\mathcal{R}$ is to select implicates that have common literals with the query. Intuitively, since we know *a priori* the vocabulary of the queries, we should only deduce consequences that have these vocabulary. One application for such paradigm is the notion of *compile on-demand* in the framework of the CMS. Starting with a set of knowledge Σ, each request for a minimal explanation of G will begin by compiling the set of G-related minimal implicates, followed by finding the resulting explanation for G. If the G-related minimal implicates are already present, the compiler will use minimal effort to detect it and produce no new G-related minimal implicates.

Thus over a sequence of queries, the set of minimal implicates will grow. If the vocabulary of all the queries covers the vocabulary of the knowledge base, then the entire knowledge base is compiled. This has the flavour of *any-time* strategy in computation as in the more queries with different vocabulary being asked, the more minimal implicates are being compiled.

If some of the G-related minimal implicates are removed because of limited storage, the compilation will detect it and regenerate those that are missing. The caveat is that the original set of knowledge must be intact at all times, that is none of the clauses it contains can be deleted[2].

We shall first introduce the notion of a *query-based* implicate. Let Σ be a formula and $Q = \{q_1, \ldots q_n\}$ be a set of distinguished literals that occur in the queries.

Definition 2.1 *A clause C is a* **query-based** *implicate of Σ restricted by Q if $\Sigma \models C$ and $C \cap Q \neq \emptyset$. The clause C is a* **query-based minimal** *implicate of Σ restricted by Q if there is no other implicate $C' \neq C$ of Σ such that C' subsumes C.*

Example 2.1

[2]The idea is similar to the notion of TeX fonts generated by METAFONT upon request when the desired font is not available.

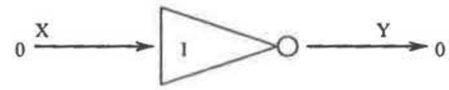

Figure 1: An Inverter Circuit I

Consider the following simple circuit with an inverter I, input X and output Y. The issue of knowledge representation for abduction (in this case diagnosis) is an on going research problem. The representation used here is consistent with earlier work in [13]. For more details on this issue, see [10]. To start, if the input is $X = 0$ and the output is $Y = 1$, the inverter I is not abnormal denoted by the predicate $\neg ab(I)$ that is,

$$X = 0 \wedge Y = 1 \rightarrow \neg ab(I).$$

Similarly, to describe the abnormality of the inverter, we need the description

$$X = 0 \wedge Y = 0 \rightarrow ab(I).$$

Also, we include the equality axioms

$$X = 0 \rightarrow \neg X = 1 \quad \text{and} \quad \neg X = 1 \rightarrow X = 0$$

to describe the fact that wire X can have exactly one value (exclusive) and wire X must have one value (existence). Thus our knowledge base Σ contains the following information:

$$\begin{array}{lll} \Sigma = \{ & \text{(i)} \ X = 0 \wedge Y = 1 \rightarrow \neg ab(I), & \text{(v)} \ X = 1 \wedge Y = 0 \rightarrow \neg ab(I), \\ & \text{(ii)} \ X = 0 \wedge Y = 0 \rightarrow ab(I), & \text{(vi)} \ X = 1 \wedge Y = 1 \rightarrow ab(I), \\ & \text{(iii)} \ X = 0 \rightarrow \neg X = 1, & \text{(vii)} \ \neg X = 1 \rightarrow X = 0, \\ & \text{(iv)} \ Y = 0 \rightarrow \neg Y = 1, & \text{(viii)} \neg Y = 1 \rightarrow Y = 0\}. \end{array}$$

The observation for the inverter circuit in example 2.1 is $X = 0 \rightarrow Y = 0$. Thus, by simplifying into clausal form, the set of query literals is $Q = \{\neg X = 0, \quad Y = 0\}$. As a notation, we shall use $MI_{|Q|}(\Sigma)$ to denote the set of all *query-based* minimal implicates of Σ restricted by Q and $MI(\Sigma)$ to denote the set of all minimal implicates of Σ without restriction. The *query-based* minimal implicates of Σ restricted by Q for the circuit example is denoted in the following set. Note that the first four are implicates from the original set Σ and the rest are deduced implicates.

$MI_{|Q|}(\Sigma) = \{$

(1) $\neg X = 0 \vee \neg Y = 1 \vee \neg ab(I)$,
(2) $\neg X = 0 \vee \neg Y = 0 \vee ab(I)$,

(3) $\neg X = 0 \vee \neg X = 1,$
(4) $Y = 1 \vee Y = 0,$

(5) $\neg X = 1 \vee ab(I) \vee Y = 0,$
(6) $X = 1 \vee Y = 0 \vee \neg ab(I),$
(7) $X = 0 \vee ab(I) \vee Y = 0,$
(8) $\neg X = 0 \vee Y = 0 \vee \neg ab(I),$
(9) $Y = 1 \vee \neg X = 0 \vee ab(I),$
(10) $X = 0 \vee \neg X = 0,$
(11) $Y = 0 \vee \neg Y = 0\}.$

The following set denotes some *non-query-based* minimal implicates of Σ restricted by Q for the circuit example but are minimal without restriction. That is, they are members of $MI(\Sigma)$.

$MI(\Sigma) - MI_{|Q|}(\Sigma) = \{$

$\neg ab(I) \vee X = 0 \vee Y = 1,$
$\neg Y = 1 \vee ab(I) \vee X = 0,$
$ab(I) \vee X = 1 \vee Y = 1,$
$X = 1 \vee \neg X = 1,$
$Y = 1 \vee \neg Y = 1, \ldots\ldots\}.$

Also, we shall use $\Sigma \models_{|Q|} C$ to denote the restricted entailment in definition 2.1 that is, $\Sigma \models C$ and $C \cap Q \neq \emptyset$ for some set of query literals Q. The following lemma 2.1 describes a property of this restricted entailment.

Lemma 2.1 *Given a formula Σ, a set of query literals Q and a clause $C \subseteq Q$, $\Sigma \models_{|Q|} C$ iff there is a $C' \in MI_{|Q|}(\Sigma)$ that subsumes C.*

Proof :

if: Let $\Sigma \models_{|Q|} C$. If there is no proper subset C' of C such that $\Sigma \models C'$, then by minimality $C \in MI_{|Q|}(\Sigma)$. Assume there is such a proper subset of C. Since $C \subseteq Q$ hence $C' \subseteq Q$. Then by the definition of minimal *query-based* implicate (definition 2.1), the smallest such subset C' is in $MI_{|Q|}(\Sigma)$.

only-if: Let $C' \in MI_{|Q|}(\Sigma)$ and C' subsumes C. By definition 2.1, $\Sigma \models C'$. Since $C' \rightarrow C$ and $C \subset Q$, therefore $\Sigma \models_{|Q|} C$. □

Other trivial facts concerning minimal implicates and *query-based* minimal implicates are:

1. $MI(\Sigma) \subseteq Th(\Sigma)$ and

2. $MI_{|Q|}(\Sigma) \subseteq Th(\Sigma)$.

Alternately, the set $MI_{|Q|}(\Sigma)$ can be defined in terms of $Th(\Sigma)$ as follows: let $Th(\Sigma)$ be the set of all theorems of Σ, thus

$$MI(\Sigma) = SUB(\ Th(\Sigma)\ )$$

and the set of *query-based* minimal implicates restricted by Q with respect to Σ is

$$MI_{|Q|}(\Sigma) = \{P | P \in MI(\Sigma), P \cap Q \neq \emptyset\}.$$

Proposition 2.1 *Let Σ be a formula, $MI(\Sigma)$ be the set of minimal implicates of Σ and $MI_{|Q|}(\Sigma)$ be the set of query-based minimal implicates of Σ restricted by Q, the set $MI_{|Q|}(\Sigma) \subseteq MI(\Sigma)$.*

Proof : Let $C \in MI_{|Q|}(\Sigma)$ and assume that $C \notin MI(\Sigma)$. By definition 2.1 the clause C is an implicate and it is minimal with respect to Σ. According to the definition of minimal implicate, C is also a minimal implicate, contradicting the assumption. Therefore $C \in MI(\Sigma)$. □

The inclusion properties of the sets $Th(\Sigma)$, $MI(\Sigma)$ and $MI_{|Q|}(\Sigma)$ are illustrated in figure 2.

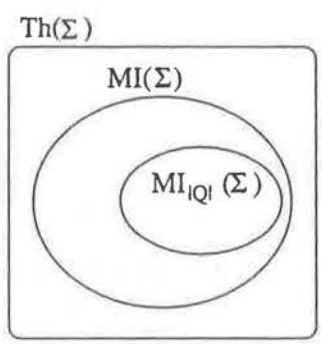

Figure 2: Set Inclusion Properties of Th, MI and $MI_{|Q|}$ of Σ.

3 Approximating Query-based Implicates

According to the definition 2.1 of *query-based* implicate, a *query-based* implicate is minimal if no proper subset of it is an implicate of Σ. This implies that a minimal *query-based* implicate relies on the fact that all other minimal implicates (without restriction) of Σ are known.

Example 3.1 *To illustrate this fact, let the query literals $Q = \{q\}$,*

$$\Sigma = \{\neg a \vee b_1, \neg b_1 \vee b_2, \ldots, \neg b_{n-1} \vee b_n, \neg b_n, \ \ \neg p \vee q, p \vee \neg a\}$$

and two deduction sequences using clauses in Σ tabulated as follows:

non-query-based implicate	query-based implicate
$\neg a \vee b_1$	$\neg p \vee q$
$\neg b_1 \vee b_2$	$p \vee \neg a$
$\neg b_2 \vee b_3$	———
$\vdots$	$\neg a \vee q$
$\neg b_{n-1} \vee b_n$	
$\neg b_n$	
———	
$\neg a$	

The deduced clause $\neg a$ from the left deduction sequence, is a *non-query-based* implicate and it subsumes $\neg a \vee q$ which is a *query-based* implicate deduced from the right deduction sequence. Thus, to ensure the minimality of $\neg a \vee q$, the long sequence of deduction on the left is required. Intuitively, a clause is minimal with respect to Σ if we know that *all* other clauses in Σ are not minimal than it. Also note that every clause on the left deduction sequence is not a *query-based* implicate because they have empty intersection with Q.

Consequently, to ensure a *query-based* implicate is minimal, all other implicates (*query-based* or not) must be generated. This poses the question can we approximate the set $MI_{|Q|}(\Sigma)$ without computing *all* the minimal implicates ? We shall attempt to characterize such an approximation with the help of a variant of query-based implicate.

Definition 3.1 *A clause C is a* **qb-implicate** *of Σ restricted by Q if $\Sigma \models C$ and $C \cap Q \neq \emptyset$. The clause C is a* **qb-minimal implicate** *of Σ restricted by Q if there is no* **qb-implicate** *$C' \neq C$ of Σ such that C' subsumes C.*

The above definition 3.1, which is a variant of *query-based* implicate, suggests that the minimality of a *qb-implicate* is restricted only to the set of *qb-implicates*. For example, let

$$\Sigma = \{ \ (1) \ \neg a \vee b, \ (2) \ \neg b, \ (3) \ \neg p \vee q, \ (4) \ p \vee \neg a, \ (5) \ h \vee c \vee q, \ (6) \ \neg h \vee \neg a\}$$

and the set of query literals $Q = \{q\}$.

$(1)\&(2) \Longrightarrow \neg a,$

$(3)\&(4) \Longrightarrow q \vee \neg a,$

$(5)\&(6) \Longrightarrow c \vee q \vee \neg a.$

Clauses $(1)\&(2)$ resolved to produce a clause which is not a *qb-implicate* because it does not contain q. The resulting resolvent of clauses $(3)\&(4)$ is a *qb-minimal implicate* because it is not subsumed by any other *qb-implicate*, and the resolvent of clauses $(5)\&(6)$ is *not* a *qb-minimal implicate* because it is subsumed by the resolvent of clauses $(3)\&(4)$. In the circuit example 2.1, clause (i) and (ii) in Σ produce the clause $\neg X = 0 \vee \neg Y = 0 \vee \neg Y = 1$ which is a *qb-minimal implicate* because it is *not* subsumed by clause (iv) in Σ which is *not* a *qb-implicate*. Intuitively, the set of all *qb-minimal* implicates $MI'_{|Q|}(\Sigma)$ contains the set $MI_{|Q|}(\Sigma)$.

Proposition 3.1 *Let Σ be a formula and Q be a set of query literals. The set $MI_{|Q|}(\Sigma) \subseteq MI'_{|Q|}(\Sigma)$.*

Proof : Let the clause $C \in MI_{|Q|}(\Sigma)$ and assume that $C \notin MI'_{|Q|}(\Sigma)$. By the definition 2.1 of *query-based* implicate, the clause C is an implicate of Σ, $C \cap Q \neq \emptyset$ and there is no $C' \subset C$ is an implicate of Σ. By the definition 3.1 of *qb-implicate*, C is a *qb-minimal implicate* of Σ, contradicting the assumption. Therefore $C \in MI'_{|Q|}(\Sigma)$. □

Since the set $MI_{|Q|}(\Sigma) \subseteq MI(\Sigma)$, trivially the intersection of $MI(\Sigma)$ and $MI'_{|Q|}(\Sigma)$ is the set $MI_{|Q|}(\Sigma)$ as shown in the following figure 3.

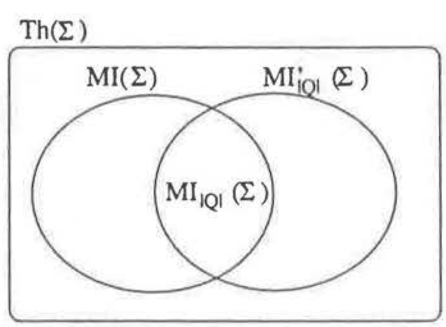

Figure 3: Set Inclusions Property of Th, MI, $MI_{|Q|}$ and $MI'_{|Q|}$ of Σ.

Proposition 3.2 *Let Σ be a set of clauses and Q be a set of query literals. The set $MI_{|Q|}(\Sigma) = MI(\Sigma) \cap MI'_{|Q|}(\Sigma)$.*

Proof : Assume that the clause $C \in MI(\Sigma) \cap MI'_{|Q|}(\Sigma)$ and $C \notin MI_{|Q|}(\Sigma)$. By the definition 3.1 of *qb-implicate*, $C \in MI'_{|Q|}(\Sigma)$ implies that C is an implicate of Σ, $C \cap Q \neq \emptyset$ and there is no $C' \subset C$ such that $C' \cap Q \neq \emptyset$ is an implicate of Σ. Since C is also in $MI_{|Q|}(\Sigma)$, no $C' \subset C$ is an implicate of Σ. By the definition 2.1 of *query-based* implicate, $C \in MI_{|Q|}(\Sigma)$ contradicting the assumption. Let the clause $C \in MI_{|Q|}(\Sigma)$. By propositions 2.1 and 3.1, $C \in MI(\Sigma) \cap MI'_{|Q|}(\Sigma)$. □

In fact, definition 3.1 suggests a systematic algorithm to compute the set $MI'_{|Q|}(\Sigma)$. To implement this, we need a notion called the consensus operation used commonly in switching theory [1]. The consensus method is simply propositional resolution in clausal form with added restrictions.

Definition 3.2 *Let $A = \{x\} \cup A'$ and $B = \{\overline{x}\} \cup B'$ be two clauses. The* **consensus** *of A and B with respect to the variable x is $CS(A, B, x) = A' \cup B'$ if $A' \cup B'$ is fundamental.*

For instance, using clause (i) and (vii) in the circuit example,

$$CS(\neg X = 0 \vee \neg Y = 1 \vee \neg ab(I),\ \ X = 1 \vee X = 0,\ \ \neg X = 0) \ = \ X = 1 \vee \neg Y = 1 \vee \neg ab(I).$$

Subsequently, using clause (viii) to resolve with the above consensus, we obtained

$$CS(X = 1 \vee \neg Y = 1 \vee \neg ab(I),\ \ Y = 1 \vee Y = 0,\ \ \neg Y = 1) \ = \ X = 1 \vee Y = 0 \vee \neg ab(I).$$

Thus, every consensus obtained from clauses in Σ is an implicate of Σ [3]. In order to obtain a *qb-implicate*, the consensus operation is given the additional restriction that $CS(A, B, x) \cap Q \neq \emptyset$ for some set of query literals Q. Obviously, $CS(A, B, x) \cap Q = \emptyset$ if both $A \cap Q = \emptyset$ and $B \cap Q = \emptyset$. This suggests that the consensus operation for finding *qb-implicates* is guided by the participating clauses that have non-empty intersection with the set of query literals.

Unfortunately, the definition of $MI'_{|Q|}(\Sigma)$ in definition 3.1 contains more implicates than those generated solely from the restricted consensus operation. For example, if $\Sigma = \{a \vee b\}$ and $Q = \{q_1, \ldots, q_n\}$, then every $a \vee b \vee q_i$ for $1 \leq i \leq n$, is a *qb-implicate* of Σ. This also implies that the size of $MI'_{|Q|}(\Sigma)$ is as large as $MI(\Sigma)$.

However, if we restrict our implicate generation to only those generated by the restricted consensus operation, we should obtain a subset of $MI'_{|Q|}(\Sigma)$ which is much smaller than $MI(\Sigma)$ and

[3]For more details, see [8].

$MI'_{|Q|}(\Sigma)$ but contains $MI_{|Q|}(\Sigma)$. To characterize such a subset, consider first the definition of a restricted *generalized consensus.*

Definition 3.3 (Restricted Generalized Consensus)
Let $C = \{C_1, \ldots, C_n\}$ and $Q = \{q_1, \ldots, q_m\}$. A **generalized consensus** *of C restricted by Q is defined as*

$$GCS(C) = \begin{cases} C_1 & \text{if } C = \{C_1\} \text{ and } C_1 \cap Q \neq \emptyset \\ S = CS(\ GCS(\{C_1, \ldots, C_{n-1}\}),\ C_n,\ x\ ) & \text{if } C = \{C_1, \ldots, C_n\} \text{ and } S \cap Q \neq \emptyset \\ \textit{undefined} & \textit{otherwise} \end{cases}$$

Note that the definition implies the order of clauses in C is crucial because, if rearranged, it might not yield a restricted generalized consensus. That is when $|C| > 2$ it is not commutative. For example, let $C = \{(1)\ \neg a \vee b,\ (2)\ \neg b \vee c,\ (3)\ \neg c \vee q\}$ and $Q = \{q\}$.

1. $GCS(\{1,2,3\}) =$ *undefined* because $CS(1,2,b) \cap Q = \emptyset$.

2. $GCS(\{1,3,2\}) =$ *undefined* because there is no consensus between clause (1) and (3).

3. $$\begin{aligned} GCS(\{3,2,1\}) &= CS(\ CS(3,2,c),\ \ 1,b) \\ &= CS(\ CS(\ \neg c \vee q,\ \ \neg b \vee c,\ \ c),\ \ \neg a \vee b,\ \ b) \\ &= CS(\ \neg b \vee q,\ \ \neg a \vee b,\ \ b) \\ &= \neg a \vee q \end{aligned}$$

Fortunately, if a generalized consensus exists for a set of clauses C restricted by Q, then any permutation of C that has a generalized consensus restricted by Q will yield the same consensus.

Proposition 3.3 *Let $C = \{C_1, \ldots, C_n\}$ be a set of clauses and $GCS(C)$ be a generalized consensus restricted by Q for the ordering in C. Any permutation C' of C such that $GCS(C')$ exists, then $GCS(C') = GCS(C)$.*

Proof : Observe that each consensus operation $CS(A, B, x)$ involves removing a pair of complimentary literals, say $\{x, \neg x\}$. Thus the $GCS(C) = S$ has all the pairs of complimentary literals $\{x_1, \neg x_1\}, \ldots, \{x_{n-1}, \neg x_{n-1}\}$ removed by the sequence of consensus operations

$$CS(C_1, C_2, x_1), \quad CS(CS(C_1, C_2, x_1), C_3, x_2), \quad \ldots, \quad CS(GCS(\{C_1, \ldots, C_{n-1}\}), C_n, x_{n-1})$$

to yield S. Any permutation $\mathcal{C}'$ of $\mathcal{C}$ such that $GCS(\mathcal{C}') = S'$, if the same set of complimentary literals $\{x_1, \neg x_1\}, \ldots, \{x_{n-1}, \neg x_{n-1}\}$ is removed, then $S' = S$. If $S' \neq S$, then there is at least a pair of complimentary literals $\{y, \neg y\}$, $y \neq x_j, 1 \leq j \leq n-1$ used in place of some $\{x_i, \neg x_i\}$. Since every clause $C_k, 1 \leq k \leq n$ participates in the restricted generalized consensus operation to produce S', thus $\{x_i, \neg x_i\} \in S'$ contradicting the definition that $GCS(\mathcal{C}') = S'$ is fundamental. □

As a consequence of the above proposition, the set of all possible consensus of a set of clauses $\mathcal{C}$ restricted by Q is the set of all possible generalized consensus of the members of the powerset of $\mathcal{C}$ restricted by Q. Additionally, we shall also speak of the minimality of a consensus in terms of the subsumption relation.

Corollary 3.1 *Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ be a set of clauses. The set of all minimal consensus of $\mathcal{C}$ restricted by Q is $\mathcal{G}(\mathcal{C}) = SUB(\{GCS(S) \mid \forall S \subseteq \mathcal{C}\})$.*

Finally, the set inclusion properties amongst $\mathcal{G}(\Sigma)$, $MI_{|Q|}(\Sigma)$ and $MI'_{|Q|}(\Sigma)$ are expressed in the following lemma.

Lemma 3.1 *Let Σ be a formula, Q be a set of query literals and $\mathcal{G}(\Sigma)$ be the set of all minimal consensus restricted by Q.*

1. *$\mathcal{G}(\Sigma) \subseteq MI'_{|Q|}(\Sigma)$ and*
2. *$MI_{|Q|}(\Sigma) \subseteq \mathcal{G}(\Sigma)$.*

Proof : (1)$\mathcal{G}(\Sigma) \subseteq MI'_{|Q|}(\Sigma)$: Assume that there is an $S \in \mathcal{G}(\Sigma)$ such that $S \notin MI'_{|Q|}(\Sigma)$. Observe that every minimal consensus restricted by Q is an implicate of Σ. Thus S is a *qb-implicate* and since $S \in \mathcal{G}(\Sigma)$ implies that no proper subset of S is in $\mathcal{G}(\Sigma)$, hence S is a minimal *qb-implicate* (definition 3.1) contradicting $S \notin MI'_{|Q|}(\Sigma)$.

(2)$MI_{|Q|}(\Sigma) \subseteq \mathcal{G}(\Sigma)$: Let $S \in MI_{|Q|}(\Sigma)$. By the definition 2.1, S is an implicate of Σ, S is minimal and $S \cap Q \neq \emptyset$. By corollary 3.1,the set $\mathcal{G}(\Sigma)$ contains all the implicates that are minimal and has non-empty intersection with Q, therefore $S \in \mathcal{G}(\Sigma)$. □

Figure 4 illustrates the set inclusion relationship amongst $Th(\Sigma)$, $MI(\Sigma)$, $MI_{|Q|}(\Sigma)$, $MI'_{|Q|}(\Sigma)$ and the approximated implicates $\mathcal{G}(\Sigma)$ denoted by the circular dotted line. The partition in the dotted circle ($\mathcal{Z}$) that is outside of $MI_{|Q|}(\Sigma)$ contains those *qb-minimal implicates* that are not minimal globally with respect to Σ. As shown in example 3.1, the $\neg a \vee q$ is a *qb-implicate* generated by the restricted generalized consensus operation but is not minimal with respect to Σ globally.

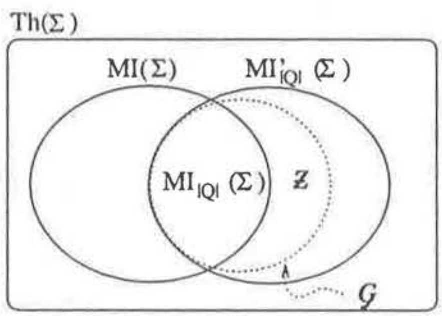

Figure 4: Set Inclusions Property of $Th, MI, MI_{|Q|}, MI'_{|Q|}$ and the approximated implicates $\mathcal{G}$ of Σ.

In fact, we can characterize the property of the set $\mathcal{Z}$ exactly. First, it is obviousthat $\mathcal{G}(\Sigma) = MI_{|Q|}(\Sigma) \cup \mathcal{Z}$.

Corollary 3.2 *Let Σ be a formula, Q a set of query literals and a clause Z. If $Z \in \mathcal{Z}$ then there exists a clause $M \in MI(\Sigma)$ such that $M \cap Q = \emptyset$ and M subsumes Z.*

Proof : Assume that $Z \in \mathcal{Z}$ thus $Z \in \mathcal{G}(\Sigma)$. Since Z is a *qb-implicate* and it is not minimal by the fact that it is not in $MI_{|Q|}(\Sigma)$, therefore there is an implicate that subsumes it. If there is an implicate $M \in MI_{|Q|}(\Sigma)$ that subsumes it, then by the definition of $\mathcal{G}(\Sigma)$, $Z \notin \mathcal{G}(\Sigma)$ contradicting the fact that $Z \in \mathcal{G}(\Sigma)$. Thus, the only minimal implicate M that can subsumes Z is the minimal implicate outside of $MI_{|Q|}(\Sigma)$. By minimality, such $M \in MI(\Sigma)$ and $M \cap Q = \emptyset$. □

4 The Algorithm

Intuitively, the restricted generalized consensus suggests that the search for a *qb-implicate* through a chain of restricted consensus operations should always select participating clauses that has a non-empty intersection with Q. This fact is stated more precisely in corollary 4.1.

Corollary 4.1 *Let Σ be a formula and Q a set of query literals. If $\mathcal{S} \subseteq \Sigma$ such that $\forall S_i \in \mathcal{S}(S_i \cap Q = \emptyset)$, then $GCS(\mathcal{S}') =$ undefined for all $\mathcal{S}' \subseteq \mathcal{S}$.*

That is, any subset $\mathcal{S}$ of Σ such that every member of it has an empty intersection with Q, we can safely ignore for purposes of restricted consensus. Recall that in the definition of restricted generalized consensus (definition 3.3), the GCS of a set of clauses can be computed via a sequence of restricted consensus of two clauses. Thus by searching for two clauses at a time such that at least one of the clauses have non-empty intersection with Q ensures the elimination of redundant search as suggested by corollary 4.1. We shall introduce a naive method for computing restricted consensus in algorithm 1.

Algorithm 1 *Naive QBIG*

Input: A formula Σ and a set of query literals Q.
Output: The set $\mathcal{G}(\Sigma)$.

Let $temp \leftarrow \emptyset$
While $temp \neq \Sigma$ do
    $temp \leftarrow \Sigma$
    $\Sigma \leftarrow \Sigma \cup CS(A, B, x)$ for some $A, B \in \Sigma$ such that $A \cap Q \neq \emptyset$ or $B \cap Q \neq \emptyset$.
end
$\Sigma \leftarrow \Sigma \cup \{ \{q, \neg q\} \mid \forall q \in Q\}$
$\Sigma \leftarrow$ qb-Subsumption(Σ, Q)

end

Algorithm 1: Naive QBIG

Finally, to obtain the subset of *qb-minimal* implicates, the subsumption operation is performed according to the rule that clause A subsumes clause B only if $A \subset B$ and both clauses have non-empty intersection with Q. Algorithm 2 shows a simple restricted subsumption method.

To illustrate the search space, consider the following set of clauses in Σ and assume that the query literals are $Q = \{q_1, q_2\}$.

$$
\begin{array}{lcccccc}
(1) & & & & a_4 & \rightarrow & c \\
(2) & a_1 & \wedge & a_2, & c & \rightarrow & d \\
(3) & a_3 & \wedge & a_1, & d & \rightarrow & q_1 \\
(4) & a_4 & \wedge & a_2, & d & \rightarrow & q_2
\end{array}
$$

Algorithm 2 *qb-Subsumption*

Input: A formula Σ and a set of query literals Q.
Output: Minimal clauses of Σ restricted by Q.

Let $temp \leftarrow \emptyset$
While $temp \neq \Sigma$ do
  $temp \leftarrow \Sigma$
  $\Sigma \leftarrow \Sigma - A$ for some $A \in \Sigma$ such that $\exists B \in \Sigma\, (B \subset A)$, $A \cap Q \neq \emptyset$ and $B \cap Q \neq \emptyset$.
end

end

Algorithm 2: Algorithm qb-Subsumption

The set of minimal implicates are depicted in figure 5. The original clauses are enclosed in solid boxes and the derived minimal implicates are outlined by dotted boxes. A solid arrow between boxes indicates potential consensus and a dotted arrow denotes subsumption relation. The numbers in brackets denote the clauses involved in generating the implicate. For example, box $(2,3)$ is the consensus of clauses (2) and (3). Notice that the dotted box $(1,2)$ is not a *qb-implicate* because it has no common literals with Q. Also note that clause $(1,2)$ is also the intermediate clause that produces the other two implicates denoted by dotted boxes $(1,2,3)$ and $(1,2,4)$ respectively.

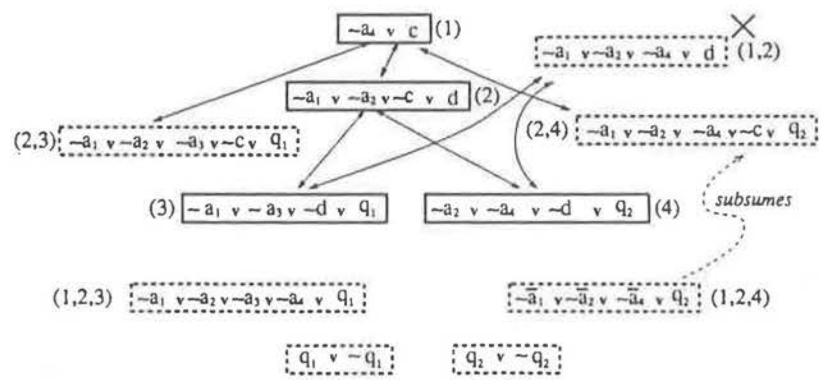

Figure 5: Minimal Implicates Space

Alternatively, the other two implicates $(2,3)$ and $(2,4)$, which are *qb-implicates*, also are capable of producing the *qb-implicates* $(1,2,3)$ and $(1,2,4)$ respectively. Thus, following the algorithm QBIG, we first select a clause that has non-empty intersection with Q, that is either clause (3) or (4). Let us assume clause (3) is selected and resolved with clause (2) with respect to literal $\neg d$ to produce clause $(2,3)$. Subsequently, clause $(2,3)$ resolves with clause (1) with respect to literal $\neg c$ to produce clause $(1,2,3)$.

Similarly, when clause (4) is chosen to resolve with clause (2) with respect to literal $\neg d$, it yields clause $(2,4)$. A subsequent consensus operation with clause (1) with respect to literal $\neg c$ produces clause $(1,2,4)$. Notice that clause $(1,2)$ is never produced using algorithm QBIG and thus the computational saving.

Since the complexity of minimal implicates generation is equivalent to the number of minimal implicates, the restriction on generating only a subset of minimal implicates obviously suggests lower complexity. Nevertheless the worst case where every clause in Σ has non-empty intersection with Q, will have the same complexity as the non-restricted case. The key argument is that it provides a framework for *query-based* computation as in the *compile on-demand* scenario. Obviously the above algorithm is naive and further improvement can be achieved. We shall leave this investigation for future work.

5 Approximated Explanations

In this section, we shall demonstrate the use of the approximated *query-based* minimal implicates in computing explanations. Recall that an explanation for a query G with respect to a formula Σ is (1) $\Sigma \models E \rightarrow G$ and (2) $\Sigma \not\models \overline{E}$. And E is minimal if no proper subset of it has the same property. To compute the set of minimal explanations from $MI(\Sigma)$, one simply computes the set [13]

$$ME(G, MI(\Sigma)) = SUB(\{E \mid M \in MI(\Sigma), M \cap G \neq \emptyset \text{ and } \overline{E} = M - G\}).$$

Recall our motivation is to compute the set of minimal explanations from a smaller set of minimal implicates $MI_{|Q|}(\Sigma)$, or more precisely the approximated set $\mathcal{G}(\Sigma)$ computed using algorithm QBIG. The question remains what are the discrepancies between the minimal explanations obtained from $MI(\Sigma)$ and $MI_{|Q|}(\Sigma)$ and thus $\mathcal{G}(\Sigma)$. In reality, what is important to us is the property of explanation obtained from the set $\mathcal{G}(\Sigma)$. Since the set $\mathcal{G}(\Sigma)$ is a superset of $MI_{|Q|}(\Sigma)$, thus studying the properties of explanations from $MI_{|Q|}(\Sigma)$ will suggest to the properties of explanations obtained from the set $\mathcal{G}(\Sigma)$.

There are two basic properties for an explanation E of G with respect to a knowledge base Σ. These are the *minimality* and the *consistency* of the explanation E. Let the set $ME(G, MI_{|Q|}(\Sigma))$ be the set of all minimal explanations for G with respect to Σ restricted by the set of query literals Q, that is

$$ME(G, MI_{|Q|}(\Sigma)) = SUB(\{E \mid M \in MI_{|Q|}(\Sigma), M \cap G \neq \emptyset \text{ and } \overline{E} = M - G\}).$$

Note that in the general case, there is no restriction that $G \cap Q \neq \emptyset$. This is deliberate since this framework is also aimed at other potential application like assumption based reasoning [9].

Corollary 5.1 *Let Σ be a formula, Q be a set of query literals and G be a non-empty clause. If $E \in ME(G, MI_{|Q|}(\Sigma))$ then E is consistent with Σ.*

Proof : If $E \in ME(G, MI_{|Q|}(\Sigma))$, then $\Sigma \models \overline{E} \cup G$ for non-empty G and $\overline{E} \cup G$ is minimal. Assume that $\Sigma \models \overline{E}$, but $\overline{E} \subset (\overline{E} \cup G)$ thus contradicting $\overline{E} \cup G$ is minimal. Consequently $\Sigma \not\models \overline{E}$ or $\Sigma \cup E$ is consistent. □

Furthermore, if $E \in ME(G, MI_{|Q|}(\Sigma))$ then E is minimal for G with respect to $MI_{|Q|}(\Sigma)$. Unfortunately, E is no longer minimal with respect to Σ globally. More precisely,

Corollary 5.2 *Let Σ be a formula, Q be a set of query literals and G be a non-empty clause. If $E \in ME(G, MI_{|Q|}(\Sigma))$ then there exists an $E' \in ME(G, MI(\Sigma))$ such that E' subsumes E.*

Proof : Let $E \in ME(G, MI_{|Q|}(\Sigma))$, thus by the definition of $ME(G, MI_{|Q|}(\Sigma))$, there exists $M \in MI_{|Q|}(\Sigma)$ such that $\overline{E} = M - G$ and $M \cap G \neq \emptyset$. Since M is minimal with respect to Σ, therefore $M \in MI(\Sigma)$ and $\overline{E} = M - G$. By the minimality of $ME(G, MI(\Sigma))$, there is an $E' \in ME(G, MI(\Sigma))$ that subsumes E. □

Note that the E' does not necessary equal to E as illustrated in the following example. Let $\Sigma = \{a \vee b \vee c,\ \ a \vee d \vee c \vee q\}$, $Q = \{q\}$ and $G = b \vee c \vee q$. Thus, $a \vee d \vee c \vee q \in MI_{|Q|}(\Sigma)$ and both $a \vee b \vee c$ and $a \vee d \vee c \vee q$ are in $MI(\Sigma)$. From $MI_{|Q|}(\Sigma)$ the explanation $\neg a \vee \neg d \in ME(G, MI_{|Q|}(\Sigma))$. Conversely, from $MI(\Sigma)$, $\neg a \in ME(G, MI(\Sigma))$ because $\neg a$ subsumes the explanation $\neg a \vee \neg d$. Thus the explanation $\neg a \vee \neg d$ in $ME(G, MI_{|Q|}(\Sigma))$ is not in $ME(G, MI(\Sigma))$. This is simply stating the fact that a minimal explanation obtained from a smaller subset $MI_{|Q|}(\Sigma)$ cannot guarantee that it is minimal with respect to explanations obtained from a larger set $MI(\Sigma)$.

Obviously, the converse of corollary 5.2 does not hold in general either. For example, let $\Sigma = \{a \vee b \vee c,\ \ a \vee q \vee c\}$, the set of query literals $Q = \{q\}$ and $G = b \vee c$. Since $a \vee q \vee c \in MI_{|Q|}(\Sigma)$

hence a minimal explanation of G from $MI_{|Q|}(\Sigma)$ is $E = \neg a \wedge \neg q$. That is, the clause $a \vee q \vee c$ is a *query-based* minimal implicate and E is obtained from $MI_{|Q|}(\Sigma)$. Conversely, $a \vee b \vee cMI_{|Q|}(\Sigma)$ but $a \vee b \vee cMI(\Sigma)$ and hence a minimal explanation for G with respect to Σ is $E' = \neg a$. That is, the clause $a \vee b \vee c$ is a minimal implicate but not *query-based.* Now that E' subsumes E thus E is not a minimal explanation for G globally with respect to Σ.

Fortunately, there is a special case when the query G is a unit clause (i.e. contains a single literal) and $G \cap Q \neq \emptyset$, these two sets are equivalent as expressed in the following lemma 5.1.

Lemma 5.1 *Let Σ be a formula, Q be a set of query literals and G is a unit clause such that $G \cap Q \neq \emptyset$. The set $ME(G, MI_{|Q|}(\Sigma)) = ME(G, MI(\Sigma))$.*

Proof : Let $E \in ME(G, MI_{|Q|}(\Sigma))$ and since G is a unit clause, $(\overline{E} \cup G) \in MI_{|Q|}(\Sigma)$. By corollary 5.2, there is an $E' \in ME(G, MI(\Sigma))$ that subsumes E. Assume that $E' \subset E$, then $\Sigma \models \overline{E'} \cup G$ and $(\overline{E'} \cup G) \cap Q \neq \emptyset$. But $(\overline{E'} \cup G) \subset (\overline{E} \cup G)$ thus contradicting $(\overline{E} \cup G) \in MI_{|Q|}(\Sigma)$, therefore $E' = E$.

Let $E \in ME(G, MI(\Sigma))$ and since G is a unit clause, $(\overline{E} \cup G) \in MI(\Sigma)$. Since $(\overline{E} \cup G) \cap Q \neq \emptyset$ and by minimality, $(\overline{E} \cup G) \in MI_{|Q|}(\Sigma)$. Assume that there is a $E' \in ME(G, MI_{|Q|}(\Sigma))$ such that $E' \subset E$. Since G is a unit clause, $(\overline{E'} \cup G) \in MI_{|Q|}(\Sigma)$. But $\overline{E'} \subset \overline{E}$ contradicting $(\overline{E} \cup G) \in MI_{|Q|}(\Sigma)$, therefore $E \in ME(G, MI_{|Q|}(\Sigma))$. □

In summary, obtaining explanations from the set $MI_{|Q|}(\Sigma)$ preserves *consistency* and in general sacrifices *minimality* globally with respect to Σ. Nevertheless, it does preserve *minimality* with respect to $MI_{|Q|}(\Sigma)$ locally. In the special case, if the query G is a unit clause and has non-empty intersection with Q, we preserve both properties and gain by searching only a subset of the minimal implicates space.

Turning to the set of approximated *query-based* minimal implicates of $\mathcal{G}(\Sigma)$, we know from lemma 3.1 that $MI_{|Q|}(\Sigma) \subseteq \mathcal{G}(\Sigma)$. Thus, explanations obtained from the set $\mathcal{G}(\Sigma)$ inherit at least the properties mentioned above. Recall from figure 4 and corollary 3.2, there exist implicate in the set $\mathcal{G}(\Sigma)$ that are not minimal globally with respect to Σ. This is illustrated by the area denoted by $\mathcal{Z}$ in figure 4. This property implies the explanations obtained from implicates in this subset $\mathcal{Z}$ might not be consistent globally with respect to Σ.

Since the subset $\mathcal{Z}$ is not known *a priori* , the question remains whether there is a sign where the explanations obtained from the set $\mathcal{G}(\Sigma)$ can guarantee consistency. Let

$$ME(G, \mathcal{G}(\Sigma)) = SUB(\{E \mid M \in \mathcal{G}(\Sigma), M \cap G \neq \emptyset \text{ and } \overline{E} = M - G\})$$

be the set of minimal explanations obtained from the set $\mathcal{G}(\Sigma)$ computed by algorithm QBIG.

Lemma 5.2 *Let Σ be a formula, Q be a set of query literals, $\mathcal{G}(\Sigma)$ be the set of implicates computed from algorithm QBIG and G be a clause. If $E \in ME(G, \mathcal{G}(\Sigma))$ and $(\overline{E} \cup G) \subseteq Q$, then $\Sigma \cup E$ is consistent.*

Proof : Since $E \in ME(G, \mathcal{G}(\Sigma))$ and $(\overline{E} \cup G) \subseteq Q$, by the definition of $\mathcal{G}(\Sigma)$, $(\overline{E} \cup G)$ is a minimal *query-based* implicate, that is, $(\overline{E} \cup G) \in MI_{|Q|}(\Sigma)$. Assume that E is inconsistent with Σ, that is, there is a $E' \subseteq E$ such that $\Sigma \models \overline{E'}$. But $\overline{E'} \subseteq (\overline{E} \cup G)$ contradicting $(\overline{E} \cup G) \in MI_{|Q|}(\Sigma)$. Therefore there cannot be such a $\overline{E'}$ and consequently $\Sigma \cup E$ is consistent. □

In short, lemma 5.2 states that if the explanation (to be precise, the negation of the explanation) and the query G contain only query literals from Q, we are certain that the explanation is consistent. Nevertheless, to ensure minimality, we require that G is a unit clause in addition to the condition stated in lemma 5.2. This is stated more formally in lemma 5.3.

Lemma 5.3 *Let Σ be a formula, Q be a set of query literals, $\mathcal{G}(\Sigma)$ be the set of implicates computed from algorithm QBIG and G be a unit clause. $E \in ME(G, \mathcal{G}(\Sigma))$ and $(\overline{E} \cup G) \subseteq Q$ iff $E \in ME(G, MI(\Sigma))$ and $(\overline{E} \cup G) \subseteq Q$.*

Proof :

if: Since $E \in ME(G, \mathcal{G}(\Sigma))$, $(\overline{E} \cup G) \in \mathcal{G}(\Sigma)$ and is minimal. Additionally, $(\overline{E} \cup G) \subseteq Q$ implies that $(\overline{E} \cup G) \in MI_{|Q|}(\Sigma)$. Since G is a unit clause, $E \in ME(G, MI_{|Q|}(\Sigma))$ and by lemma 5.1, $E \in ME(G, MI(\Sigma))$.

onlyif: Let $E \in ME(G, MI(\Sigma))$ and since $(\overline{E} \cup G) \subseteq Q$ and G is a unit clause, by lemma 5.1, $E \in ME(G, MI_{|Q|}(\Sigma))$. Hence $(\overline{E} \cup G) \in MI_{|Q|}(\Sigma)$ and it is minimal. Since $MI_{|Q|}(\Sigma) \subseteq \mathcal{G}(\Sigma)$ by lemma 3.1, $(\overline{E} \cup G) \in \mathcal{G}(\Sigma)$. Again by minimality and the fact that G is a unit clause, $E \in ME(G, \mathcal{G}(\Sigma))$. □

Consider example 3.1 again, the algorithm QBIG will only derive the implicate $\neg a \vee q$ but not the long chain for *non-query-based* implicate $\neg a$. Thus if the query is q, then the explanation a obtained from $a \rightarrow q$ is inconsistent globally with respect to Σ. In short, sacrificing minimality in implicates introduces inconsistency in explanations in general except the two special cases in lemma 5.2 and 5.3.

Now, the tradeoff becomes that between the saving over computing all the minimal implicates, versus relaxing consistency in explanation. One may argue that consistency of an explanation is crucial and it should be required at any cost, but indicated in example 3.1, to derive $\neg a$ is expensive. On the other hand, one could argue that obtaining an explanation quickly is better. This is supported by two major reasons. Firstly, obtaining an explanation is better than not as in the example 3.1, where there will be no explanation for q if we insist on consistency. Thus from the information processing point of view, the user is left guessing which part of the explanation is inconsistent. Secondly, since reasoning is continuous, asking more queries will eventually lead to the detection of the inconsistency. For example, after obtaining the explanation a for q, asking for the explanation of $\neg a$ thus changing the set of query literals to $Q = \{\neg a\}$, will ensure the detection of $\neg a$ in the knowledge base.

This paper argues for the second proposition above, that is relaxing minimality and consistency in favour of faster computation. In a much looser argument, one can view such approximation as lazy reasoning as in working on only the easily accessible facts. As the demand for precision increases, more work is required.

6 ATMS implicates

In this section, we shall demonstrate the use of the definition of *selective* implicates in defining ATMS implicates. The purpose is merely an exercise to show the generality of the definition and that the formalization of ATMS is feasible following the original motivation of Reiter and de Kleer [12]. Let Σ be a formula, Let $\mathcal{A} = \{a_1, \ldots a_n\}$ be a set of distinguished positive literals called *assumptions*.

Definition 6.1 *A clause C is an* **ATMS**-*implicate of Σ if $\Sigma \models C$ and*

1. $C = a_1 \wedge \ldots \wedge a_m \rightarrow q$ *for some literal q,*
2. $\{a_1, \ldots, a_m\} \subseteq \mathcal{A}$ *and $q \notin \mathcal{A}$.*

A clause C is an **ATMS**-*minimal implicate of Σ if no proper subset C' of C is an* **ATMS** *implicate of Σ. A* **nogood** *is an* **ATMS** *implicate $C = a_1 \wedge \ldots \wedge a_m \rightarrow \Box$ where $\Box$ denotes empty clause.*

Thus, the definition is also an approximation of implicates and minimal implicates with the added restrictions. Obviously without restrictions on the formula Σ, the minimality of an **ATMS**-implicate with respect to Σ globally is sacrificed anolog to minimal *qb-implicate* described earlier. Consequently, such sacrifice will result in possibly inconsistent explanations.

To circumvent this drawback, de Kleer in his original ATMS [3], restricted the formula Σ to be composed of HORN-clauses. Furthermore, no *assumption* literals can be justified, that is, neither $a_1 \wedge a_2 \rightarrow a \vee b$ nor $c \rightarrow a_3$ is allowed. It can be shown easily that such a restriction guarantees that every minimal **ATMS**-implicate is minimal globally with respect to Σ [4].

The ATMS algorithm exploits all the restrictions mentioned above so that computing minimal **ATMS**-implicates is fast. Using the above definition would allow us to study variants of ATMS and their corresponding algorithms by varying the restrictions. For example, one interesting variation is to allow multiple definitions of sets of *assumption* literals $\mathcal{A}_i$, for some integer $i > 0$. Thus, for each $\mathcal{A}_i$, we build a *window* (set) of **ATMS-minimal** implicates on demand. Searching for explanations for a query can switch among *windows* depending on the focus of the sets of *assumption* literals.

7 Length-based Implicates

In this section, we describe another example of approximation using the length of a clause as its measurement.

Definition 7.1 *A clause L is a* **length-n** *implicate of Σ if $\Sigma \models L$ and $|L| \leq n$ for some natural number n. A clause L is a* **length-n** *minimal implicate of Σ if no proper subset L' of L is a* **length-n** *implicate of Σ.*

The motivation for such a definition is based on the resources bound for computing time and storage. For instance, under stringent storage requirements, requesting *length-3* minimal implicates proceeds by computing the consensus of two clauses A and B such that $|A \cup B| \leq 5$. Thus after removing the complimentary literals, the size of the consensus must be less than or equal to 3 literals. Such a constraint will avoid generating implicates of size greater than 3 and potentially save time and storage. Obviously if $n \geq |\mathcal{V}|$, the size of the vocabulary, then it is equivalent to generating all the minimal implicates. Also note that if the clause L is a *length-i* minimal implicate of Σ, then L is a minimal

[4]For non-HORN extension in ATMS, see NATMS in [4].

implicate of any length and thus Σ.

Corollary 7.1 *Let Σ be a formula and $\mathcal{L}_i$ be the set of all* **length-i** *minimal implicates. If $L \in \mathcal{L}_i$ then $L \in MI(\Sigma)$.*

Proof : Let $L \in \mathcal{L}_i$ and note that L can only be subsumed by another implicate smaller than L. By the definition of *length-based* minimal implicate (definition 7.1), the minimality of L on *length-i* ensures no other implicate of length greater than i can subsume L. Consequently L is minimal with respect to Σ and $L \in MI(\Sigma)$. $\square$

As a consequence, an explanation E for a query G generated from *length-i* minimal implicates guarantees the consistency of E with respect to Σ. Unfortunately, it does not guarantee the minimality of the explanation E for all lengths as illustrated by the following example.

Example 7.1 *Let $\Sigma = \{(1)\ a \vee b \vee q_1, (2)\ a \vee c \vee q_1 \vee q_2\}$ and $G = c \vee q_1 \vee q_2$ and $Q = \{q_1, q_2\}$. The implicate (1) has length 3 and (2) has length 4. The explanation generated from length-3 implicates is $a \vee b$ and from length-4 implicates is a, which is more minimal.*

Nevertheless, at the expense of minimality of the explanation, this idea also has the advantage of being able to *compile on-demand.* For example, given a query G, generate the explanations for G from the set of all *length-i* minimal implicates. If more precision is required, generate minimal implicates of length $i + 1$ and so on. This again, has the flavour of *any-time* computation in which the length as the variable, will guide the computation of a partial subset of minimal implicates to the complete set of minimal implicates as the length increases.

With regard to the issue of computation, generating smaller implicates first has the advantage of subsuming larger implicates. This is similar to the *unit-resolution* strategy in theorem proving where a unit literal is resolved first. Other uses for this definition include the formalization of constraint satisfaction problems in ATMS as presented in [5].

8 Conclusions

This paper has presented a general definition for an approximated implicate called the *selective* implicates. Three instances of *selective* implicates include *query-based*, ATMS and *length-based* implicates were studied. In depth study of the properties of *query-based* minimal implicates, their variant *qb-minimal* implicates and the approximated set $\mathcal{G}(\Sigma)$ generated by the algorithm QBIG suggests the feasibility of such approach. The explanation for a given query generated from the approximated subset of minimal implicates $\mathcal{G}(\Sigma)$ deviates from the explanation generated from the set of minimal implicates of Σ in two aspects. Firstly, the approximated subset of minimal implicates contains possibly globally non-minimal implicates which in turn introduces inconsistency in explanation. Secondly, the minimality of an explanation is sacrificed because not all minimal implicates are available. Exceptions to the above two discrepancies were also discovered.

An **ATMS**-implicate was defined to demonstrate the generality of the definition of *selective* implicates and finally, a *length-based* implicate was introduced to further illustrate the notion of approximating implicates. As for future work, specific algorithms based on the consensus method for generating specific *selective* implicates must be investigated. Hopefully, the classes of consensus methods will reveal the possibility of amalgamating them into a system as a whole. Further investigation into new instances of *selective* implicates based upon application domain should be rewarding.

Acknowledgement: This paper is the result of the persistent questioning on the issue of the complexity of truth maintenance systems by professor Alan Mackworth. The notion of approximation based on the query was then suggested and as a result, the definition of selective implicates. The author is also grateful to George Tsiknis, Peter Apostoli, Jane Mulligan, David Poole and the members of the ISA reasoning group for their comments and criticism.

References

[1] T.C. Bartee, I.L. Lebow, and I.S. Reed. *Theory and Design of Digital Machines.* McGraw-Hill Book, 1962.

[2] A.K. Chandra and G. Markowsky. On the Number of Prime Implicants. *Discrete Mathematics*, 24:7–11, 1978.

[3] Johan de Kleer. An Assumption-based TMS. *Artificial Intelligence*, 28, 1986.

[4] Johan de Kleer. A General Labeling Algorithm for Assumption-based Truth Maintenance. In *Proceeding of AAAI-88*, pages 188–192, 1988.

[5] Johan de Kleer. A Comparision of ATMS and CSP techniques. In *Proceeding of IJCAI-89*, pages 290–296, 1989.

[6] J. J. Finger and M. R. Genesereth. RESIDUE: A Deductive Approach to Design Synthesis. Memo HPP 1, Department of Computer Science, Stanford University, 1985. Stanford Heuristic Programming Project.

[7] Alex Kean. Partial Compilation in Clause Management Systems. Technical report, Department of Computer Science, University of British Columbia, 1991. in preparation.

[8] Alex Kean and George Tsiknis. An Incremental Method for Generating Prime Implicants/Implicates. *Journal of Symbolic Computation*, 9:185–206, 1990.

[9] Alex Kean and George Tsiknis. Assumption based Reasoning and Clause Management Systems. Technical Report 9, Department of Computer Science, University of British Columbia, 1990.

[10] David Poole. Representing Knowledge for Logic-based Diagnosis. In *Proceedings of International Conference on Fifth Generation Computing*, Tokyo, 1988.

[11] David Poole, Randy Goebel, and Romas Aleliunas. Theorist: A logical Reasoning System for Defaults and Diagnosis. CS Research Report 6, Department of Computer Science, University of Waterloo, 1986. Logic Programming and Artificial Intelligence Group.

[12] Raymond Reiter and Johan de Kleer. Foundations of Assumption-Based Truth Maintenance Systems: Preliminary Report. In *Proceeding of AAAI-87*, pages 183–188, Seatle, Washington, 1987.

[13] George Tsiknis and Alex Kean. Clause Management Systems (CMS). Technical Report 21, Department of Computer Science, University of British Columbia, 1988.